



Gráficos em Vetor para Jogos Web



lá! Que acha de trabalhar um pouco com desenhos vetoriais?

Nessa unidade voltaremos um pouco ao HTML, vendo como utilizar elementos e atributos para gerar representações gráficas. Para isso vamos estudar como fazer desenhos vetoriais com o SVG. Como isso será inserido diretamente no documento HTML, teremos todas as vantagens já vistas antes, como usar o JavaScript para manipular eventos de imagem ou usar o CSS par estilizar.

Introdução aos Gráficos Vetoriais

Objetos vetoriais são desenhos gerados a partir de fórmulas matemáticas. As representações vetoriais são criadas com vetores matemáticos combinados por ligações de pontos e segmentos num espaço bidimensional. Isso faz com que os elementos possam ser redimensionados sem perder a qualidade do traço.

Conhecidas também como desenhos ou gráficos vetoriais, as representações vetoriais são utilizadas em gráficos que devem ser visíveis em diferentes plataformas e telas (QUEIROZ et al., 2021). São úteis quando é necessário realizar modificações sem que haja perda de qualidade da imagem (como ocorre nas representações em bitmap).

O quadro abaixo mostra os principais formatos de representação vetorial:

Tipos de formatos para gráficos vetoriais	Características
SVG (Scalable Vector Graphics)	Descreve vetorialmente conteúdos estáticos, dinâmicos e animados. Muito utilizado na *web* e em dispositivos móveis.
CDR (Corel Draw)	Formato de gráfico vetorial gerado pelo programa Corel Draw, mas é reconhecido pela maioria dos editores de imagem.
AI (Adobe Illustrator)	Formato de gráfico vetorial gerado pelo programa Adobe Illustrator. Muito utilizado para o desenho de logotipos, assinaturas visuais, etc.
EPS (Encapsulated PostScript)	Formato vetorial padrão para armazenar gráficos PostScript de alta resolução. Pode ser incorporado dentro de outros documentos.

Nesta unidade, já que estamos estudando jogos na web, trabalharemos com o SVG. SVG é a abreviatura de *Scalable Vector Graphics* ou gráficos vetoriais escalonáveis. Trata-se de uma linguagem XML[1] para descrever desenhos e gráficos vetoriais bidimensionais estáticos, dinâmicos e animados. Assim como vimos sobre o HTML5, o SVG é um padrão recomendado e criado pela W3C (W3C, 2021).

Podemos usar o SVG junto com nosso programa JavaScript para manipular eventos de imagem com base no DOM, assim como podemos usar CSS para estilizá-los. Com o formato SVG vamos trabalhar com três tipos de objetos gráficos: imagens, textos ou formas geométricas vetoriais.

Usando o SVG

Dentro de um documento HTML podemos inserir uma representação gráfica em SVG por meio das tags **<svg> </svg>**. Aninhado nesta tag, podemos

inserir as formas geométricas que desejamos representar, juntamente com seus atributos HTML.



Assim, um pequeno exemplo para exibir uma representação de um círculo seria como o trecho de código HTML a seguir:

```
<!DOCTYPE html>
<html>
  <body>

    <h1>Meu primeiro SVG</h1>

    <svg width="100" height="100">
      <circle cx="50" cy="50" r="40" stroke="black" stroke-width="4" fill="red"/>
    </svg>

  </body>
</html>
```

Explicando o exemplo, temos:

- Os atributos **width** e **height** definem a largura e altura do elemento <svg>. No exemplo, teremos uma área para o desenho dada por 100 pixels de altura e largura.

- O elemento **<circle>** é usado para desenhar um círculo. Seus atributos **cx** e **cy** definem as coordenadas x e y do centro desse círculo. Se cx e cy não forem definidos, o centro do círculo é definido como (0, 0). Já o atributo **r** define o raio do círculo.

- Os atributos **stroke** e **stroke-width** controlam como o contorno aparecerá. Nesse exemplo temos o contorno do círculo como uma borda preta de 4 pixels.

- O atributo **fill** se refere à cor de preenchimento do círculo. Nesse caso, a cor de preenchimento do círculo será vermelha.

O resultado do código HTML de exemplo resultaria na imagem a seguir:





Formas básicas com SVG

O formato SVG permite o uso de objetos gráficos como imagens, textos ou formas geométricas elementares. Assim, temos vários elementos de formas pré definidos que podemos utilizar, são eles:

Elemento	Descrição
<circle>	Usado para criar círculos.
<rect>	Usado para criar retângulos e suas variações.
<elipse>	Elemento que representa elipses.
<line>	Representa uma linha reta.
<polyline>	Cria desenhos que consistem de linhas retas conectadas em vários pontos.
<polygon>	Este elemento representa um gráfico que contém pelo menos três lados.

Para fins deste resumo de unidade, vamos relacionar cada um destes elementos com seus respectivos atributos principais. Todos adaptados de BASIC (2021).

<rect>

Atributo	Descrição
x	A posição x do canto superior esquerdo do retângulo.
y	Posição y do canto superior esquerdo do retângulo.
width	A largura do retângulo.
height	Altura do retângulo.
rx	Raio x dos cantos do retângulo.
ry	Raio y dos cantos do retângulo.

Definindo os parâmetros rx e ry, o retângulo terá cantos arredondados. Se não estiverem definidos, o padrão será 0.

Então, para desenharmos um retângulo azul com largura de 250 pixels e altura dada por 100 pixels, faríamos:

```
<rect width="250" height="100" fill="blue" />
```

<elipse>

Atributo	Descrição
rx	O raio x da elipse.
ry	O raio y da elipse.
cx	A posição x do centro da elipse.
cy	A posição y do centro da elipse.

Uma <ellipse> é uma representação mais geral de <circle>. A diferença entre eles é que uma elipse tem valores dos raios x e y diferente entre si, enquanto que o círculo tem valores iguais para x e y.

Para desenharmos uma elipse amarela posicionada em (200, 80) poderíamos fazer:

```
<ellipse cx="200" cy="80" rx="100" ry="50" fill="yellow" />
```

<line>

Atributo	Descrição
x1	A posição x do ponto 1.
y1	A posição y do ponto 1.
x2	A posição x do ponto 2.
y2	A posição y do ponto 2.

Para desenharmos uma linha vermelha começando no ponto (0, 0) e terminando em (200, 200), usaríamos o código:

```
<line x1="0" y1="0" x2="200" y2="200" stroke="red"/>
```

<polyline>

Atributo	Descrição
points	Uma lista de pontos separa por espaços. Cada ponto deve ser separado [illegible]rgula e deve conter dois números representando as coordenadas x e y.

Uma <polyline> é um grupo de linhas retas conectadas. Como a lista de pontos pode ficar muito longa, todos os pontos estão incluídos em um único atributo points. Para representar 3 linhas pretas dadas pela lista de pontos (60, 40), (80, 120), (120, 140) e (200, 110), faríamos:

```
<polyline points="60,40 80,120 120,140 200,110" stroke="black" fill="none" />
```

O elemento <text>

Além dos elementos básicos de formas, é possível desenhar textos utilizando o elemento dado pela tag **<text>**. Por exemplo, para inserir o texto "*I love* SVG!" em um desenho, faríamos algo como:

```
<svg height="30" width="200">
  <text x="0" y="15" fill="red">I love SVG!</text>
</svg>
```

Seus atributos principais são:

Atributo	Descrição
x	A coordenada x do ponto inicial da linha de base do texto.
y	A coordenada y do ponto inicial da linha de base do texto.
rotate	Gira cada caractere individualmente.

A grande vantagem de utilizar o SVG em suas aplicações e interfaces de jogos é que, apesar da possibilidade de serem complexas, elas serão predominantemente leves, fáceis de animar e com possibilidade de programar interações complexas com o usuário por meio de JavaScript e CSS.

Outro fator importante é que se trata de uma representação gráfica escalável, ou seja, a imagem pode ser redimensionada sem nos preocuparmos com perda de qualidade ou distorção. Além disso, por se tratar de apenas texto, o custo de armazenamento é muito inferior assim como o de exibição, pois seu processamento se dará no navegador do usuário.

Este recurso nos permite criar experiências rápidas, qualificadas, responsivas e com ótimo desempenho. No entanto, criar estes gráficos é trabalhoso, principalmente se estivermos criando uma interface muito complexa. Para resolver isso, podemos utilizar outros softwares de desenho vetorial que exportam dados em SVG, como o InkScape que é gratuito ou softwares proprietários como o Adobe Illustrator ou o CorelDraw.

Até a próxima!

Atividade Extra

Acesse o site abaixo para desenhar objetos em SVG e para incorporá-los em seus projetos HTML, trata-se de um site gratuito para desenho vetorial. Clique no menu "view" e depois "Source..." para ver o código SVG dos seus desenhos.

- **Editor vetorial SVG**, disponível em: <https://editor.method.ac/>.

Realize o tutorial da w3schools no link abaixo para praticar desenhos com SVG. Este site tem conteúdo com exemplos de várias formas e recursos, incluindo uma lista de exemplos bem completa com formas, filtros e gradientes.

- **Tutorial SVG w3schools**, disponível em: <https://www.w3schools.com/graphics/svg_intro.asp>.

- **Exemplos "faça você mesmo" da w3schools**, disponível em: <https://www.w3schools.com/graphics/svg_examples.asp>.

Referência Bibliográfica

QUEIROZ, C.W.D. et al. **Animação Digital 2D.** Porto Alegre: Grupo A, 2021.

W3 Schools. **SVG Tutorial.** 2021. Disponível em: <https://www.w3schools.com/graphics/svg_intro.asp>.

W3C. Scalable Vector Graphics (SVG) 2021. Disponível em: <https://www.w3.org/Graphics/SVG/>.

MDN WEB DOCS. **TEXT SVG: Gráficos Vetoriais Escalonáveis.** 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/SVG/Element/text>.

MDN WEB DOCS. Basic Shapes. 2021. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Basic_Shapes>.

Ir para exercício